Partial Mutual Information Based Input Variable Selection for Supervised Learning Approaches to Voice Activity Detection

Ivan Marković[a,*], Srećko Jurić-Kavelj[a], Ivan Petrović[a]

[a] *University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, Unska 3, HR-10000, Zagreb, Croatia*

Abstract

The paper presents a novel approach for voice activity detection. The main idea behind the presented approach is to use, next to the likelihood ratio of a statistical model-based voice activity detector, a set of informative distinct features in order to, via a supervised learning approach, enhance the detection performance. The statistical model-based voice activity detector, which is chosen based on the comparison to other similar detectors in an earlier work, models the spectral envelope of the signal and we derive the likelihood ratio thereof. Furthermore, the likelihood ratio together with 70 other various features was meticulously analyzed with an input variable selection algorithm based on partial mutual information. The resulting analysis produced a 13 element reduced input vector which when compared to the full input vector did not undermine the detector performance. The evaluation is performed on a speech corpus consisting of recordings made by six different speakers, which were corrupted with three different types of noises and noise levels. In the end, we tested three different supervised learning algorithms for the task, namely, support vector machine, Boost, and artificial neural networks. The experimental analysis was performed by 10-fold cross-validation due to which threshold averaged receiver operating characteristics curves were constructed. Also, the area under the curve score and Matthew's correlation coefficient were calculated for both the three supervised learning classifiers and the statistical model-based voice activity detector. The results showed that the classifier with the reduced input vector significantly outperformed the standalone detector based on the likelihood ratio, and that among the three classifiers, Boost showed the most consistent performance.

Keywords: voice activity detection, partial mutual information, supervised learning, receiver operating characteristics curves

1. Introduction

Voice activity detection is a technique in speech processing by which presence of speech is detected in a given signal frame. This problem can be seen as a dual hypothesis problem, where a signal frame is classified as either containing speech or containing noise. In a voice activity detector (VAD), the absence of speech usually presumes presence of noise only. This system is not only of great importance for many applications, like mobile telephony, internet telephony, hearing aid devices, but also for robotics if speech oriented systems are utilized like speaker localization, speech and speaker recognition. For most of the stated research problems, it is indispensable to save on bandwidth resources by coding noise with significantly less bits, while for others it is mandatory to completely ignore frames with noise.

A VAD must provide a robust and reliable decision procedure in varying acoustical conditions. This task gets quite formidable with the varying level and type of background noise. Approaches to voice activity detection mostly differ in the type of the extracted features and in the decision models used to reach a speech/non-speech decision based on those features. A lot of attention was given to statistical model-based VADs, in which certain probabilistic properties are assumed on the coefficients of the discrete Fourier transform (DFT). For an example, in [1] they are assumed to have Gaussian distribution and this approach was further developed in [2–7] and [8]. Furthermore, special attention was given to derivation of various noise robust features and decision rules in [9, 10] and [11]. Concerning supervised learning approaches, they have been utilized in various sound processing scenarios, e.g. music classification [12], general audio signal classification (music, news, sports etc.) [13], speech inteligibility quantification [14] etc. Supervised learning based voice activity detection approaches have so far been mostly focused on applying support vector machine (SVM) by treating as features: a priori signal-to-noise ratio (SNR), a posteriori SNR and/or statistical model-based likelihood ratio [15, 16], mel frequency cepstral coefficients (MFCCs) [17], sub-band and long-term SNR [18, 19], or features used in the standard G.729B [20, 21]. Furthermore, a recent work [22] presented a novel unsupervised learning approach called support-vector-regression-based maximum margin clustering which was also tested in a voice activity detection scenario and showed comparable performance to supervised approach based on support vector machine method.

☆This work has been supported by European Community's Seventh Framework Programme under grant agreement no. 285939 (ACROSS) and the Ministry of Science, Education and Sports of the Republic of Croatia under grant No. 036-0363078-3018.

*Corresponding author

Email addresses: ivan.markovic@fer.hr (Ivan Marković), srecko.juric-kavelj@fer.hr (Srećko Jurić-Kavelj), ivan.petrovic@fer.hr (Ivan Petrović)

Our work presented in this paper surveys the supervised learning approaches to VAD and builds on upon the aforementioned related works with the following main contributions. Firstly, to the best of our knowledge, we are the first to introduce a method for input variable analysis based on partial mutual information algorithm in the context of voice activity detection. This method systematically classifies features on those that should be included and those that could be omitted from the input set, which we find extremely important when extending input spaces of supervised learning algorithms. Secondly, we extend the input space with distinct features under the hypothesis (which is tested) that this will improve the performance of VADs. While most of the features in the related works are variants on the SNR estimation (a priori, a posteriori, predicted, sub-band and long-term), with two exceptions—one which used only MFCC [17] and other which is based on features from G.729B [20], in the present paper we extended this feature space by using information from the SNR estimation in the form of a statistical-based likelihood ratio (LR) by modeling the distribution of the spectral envelope, along with several distinct features like magnitudes of some of the DFT coefficients, spectral flux, spectral centroid and bandwidth, power-normalized cepstral coefficients, MFCCs etc. Furthermore, for the classification task we present a systematic quantiative evaluation of the following three supervised learning algorithms: Boost, artificial neural networks (ANNs) and SVM, while all the related work papers on VAD utilize only SVM. The algorithms were tested and compared under varying noise conditions, namely three types of noises and three different SNRs, and showed similar performance with a slight advantage in the direction of the Boost classifier.

Although a detector can be considered as a binary classifier, for clarity throughout the paper we use the term detector to denote the statistical model-based detector based on the likelihood ratio, while the term classifier or supervised learning based VAD denotes the SVM, Boost and ANN classifiers. The rest of the paper is organized as follows. Section 2 presents the statistical model-based VADs. In Section 3, the implemented algorithms for noise spectrum estimation and *a priori* signal-to-noise ratio are presented. Section 4 presents the utilized speech corpus and evaluation metrics, while Section 5 presents the input variable selection algorithm, the input variable set and the resulting analysis. Section 6 presents the experimental evaluation of the algorithms, and Section 7 concludes the paper.

2. Statistical Model-Based Detectors

These VADs rely on statistical modeling of the DFT coefficients. All the statistical model-based VADs assume a two hypotheses scenario. Since speech is degraded with uncorrelated additive noise, the two hypotheses are as follows:

$$\begin{aligned} H_0 &: \text{speech absent} \Rightarrow \mathbf{X} = \mathbf{N} \\ H_1 &: \text{speech present} \Rightarrow \mathbf{X} = \mathbf{N} + \mathbf{S}, \end{aligned} \tag{1}$$

where the DFT coefficients of a K-point DFT of the noisy speech, noise, and clean speech are denoted as $\mathbf{X} = [X_0, X_1, \ldots, X_{K-1}]^{\mathrm{T}}$, $\mathbf{N} = [N_0, N_1, \ldots, N_{K-1}]^{\mathrm{T}}$ and $\mathbf{S} = [S_0, S_1, \ldots, S_{K-1}]^{\mathrm{T}}$, respectively.

The form of the probability density function (pdf) of $\mathbf{X}$ conditioned on the hypotheses, i.e. $p(\mathbf{X}|H_0)$ and $p(\mathbf{X}|H_1)$, depends on the distribution used to model each DFT coefficient. After the pdfs $p(\mathbf{X}|H_0)$ and $p(\mathbf{X}|H_1)$ are determined, usually a likelihood ratio on all the DFT coefficient indices k is calculated:

$$\Lambda_k = \frac{p(X_k|H_1)}{p(X_k|H_0)}, \tag{2}$$

where Λ_k becomes a vector of length K. This information is then used to calculate geometric mean which is then compared to a certain threshold in order to reach a final decision in favor of either the hypothesis H_0 or H_1:

$$\log \Lambda = \frac{1}{K} \sum_{k=1}^{K} \log \Lambda_k \underset{H_0}{\overset{H_1}{\gtrless}} \eta. \tag{3}$$

2.1. Gaussian distribution statistical model

The VAD based on Gaussian distribution was first proposed in [1], where the DFT coefficients are asymptotically independent and zero-mean complex Gaussian random variables. When both speech and noise are present, we have for each coefficient a sum of independent Gaussian variables (speech plus noise), thus resulting with a pdf of variance $\lambda_{x,k} = \lambda_{s,k} + \lambda_{n,k}$. Hence, the conditional pdfs of X_k on hypotheses H_0 and H_1 are as follows:

$$p(X_k|H_0) = \frac{1}{\pi\lambda_{n,k}} \exp\left(-\frac{|X_k|^2}{\lambda_{n,k}}\right), \tag{4}$$

$$p(X_k|H_1) = \frac{1}{\pi(\lambda_{n,k} + \lambda_{s,k})} \exp\left(-\frac{|X_k|^2}{\lambda_{n,k} + \lambda_{s,k}}\right). \tag{5}$$

Under the Gaussian distribution (GD) model, the LR is simply calculated as the ratio of (5) and (4):

$$\Lambda_k^{\mathrm{GD}} = \frac{p(X_k|H_1)}{p(X_k|H_0)} = \frac{1}{1+\xi_k} \exp\left(\frac{\gamma_k \xi_k}{1+\xi_k}\right), \tag{6}$$

where $\xi_k = \lambda_{s,k}/\lambda_{n,k}$ is the *a priori* SNR, and $\gamma_k = |X_k|^2/\lambda_{n,k}$ is the *a posteriori* SNR. A more detailed derivation can be found in [23], while the algorithms for estimation of these values are presented in Section 3.

2.2. Rayleigh and Rice distribution statistical model

In the approach proposed in [24], derived from [25], the DFT coefficients are still modelled as having a Gaussian distribution, but instead of using their joint distribution, the distribution of the signal envelope is used. The envelope of a signal, $|X_k| = \sqrt{X_{R,k}^2 + X_{I,k}^2}$, is actually the euclidean norm of the real and imaginary coefficients. Therefore, instead of looking at the distribution of the coefficients, the distribution of the signal envelope is analysed.

Under hypothesis H_0 the signal is only noise, which means that the DFT coefficients are both independent, zero-mean Gaussian variables with variance $\lambda_{n,k}/2 = \mathrm{E}[|N_k|^2]$. Under that assumption, the pdf of the euclidean distance of such DFT coefficients is a Rayleigh distribution:

$$p(X_k|H_0) = \frac{2|X_k|}{\lambda_{n,k}} \exp\left(-\frac{|X_k|^2}{\lambda_{n,k}}\right). \tag{7}$$

Under hypothesis H_1, the envelope is the euclidean norm of two independent, non-zero-mean Gaussian variables. Such pdf is a Rician:

$$\begin{aligned} p(X_k|H_1) =& \frac{2|X_k|}{\lambda_{n,k}} \exp\left(-\frac{1}{\lambda_{n,k}}\left(|X_k|^2 + |A_k|^2\right)\right) I_0\left\{\frac{2|A_k||X_k|}{\lambda_{n,k}}\right\} \\ =& \frac{2|X_k|}{\lambda_{n,k}} \exp\left\{-\frac{|X_k|^2}{\lambda_{n,k}} - \xi_k\right\} I_0\left(2\sqrt{\xi_k \frac{|X_k|^2}{\lambda_{n,k}}}\right), \end{aligned} \tag{8}$$

where A_k is the amplitude of the clean speech spectrum, $\xi_k = |A_k|^2/\lambda_{n,k}$ is the *a priori* SNR and $I_0(\cdot)$ is the modified Bessel function of the first kind and order zero. In [24] this VAD was implemented by calculating the *a posteriori* probability $p(H_1|X_k)$ of voice activity from (7) and (8) via Bayes' formula. Since in this paper the *a priori* SNR estimation, presented in Section 3, for all frequency bins is implemented, we propose the LR instead of the *a posteriori* probability $p(H_1|X_k)$. Finally, we derive the LR for Rayleigh and Rice distribution (RRD) model:

$$\Lambda_k^{\mathrm{RRD}} = \exp\{-\xi_k\}\, I_0\left(2\sqrt{\xi_k \gamma_k}\right). \tag{9}$$

In [23] we have extensively analyzed and compared the performance of three statistical model-based VADs: the GD model [1], the generalized Gaussian distribution model [5], and the RRD model [24]. The models were compared in detection performance and computational demand. On average, under three different types and levels of noises, the RRD VAD showed the best results in detection accuracy, and ranked second in computational demand. This is the reason why we chose to work further with the RRD VAD and why we try to enhance its performance with a supervised learning approach by adding, next to the LR, several other distinct features.

3. Noise Spectrum Estimation

We can see from Section 2 that the RRD VAD requires estimation of the noise spectrum $\lambda_{n,k}$ and the *a priori* SNR ξ_k. First we shall address the estimation of $\lambda_{n,k}$ and then the estimation of ξ_k.

In most VADs the noise spectrum estimation is done in a way to assume that in the first several frames only noise is present and for that time $\lambda_{n,k}$ is estimated by time averaging the spectrum of the recorded signal. Then, the VAD itself is used to discriminate between frames where speech is present and where only noise is present. When only noise is detected, $\lambda_{n,k}$ is again estimated in a time-averaging fashion.

In this paper an algorithm proposed by [26] and [27] called minima-controlled recursive averaging (MCRA) is used since it performs well in varying noise situations and it allows estimation from all frames, and not just the ones where no speech is detected.

3.1. Minima-controlled recursive averaging

As stated earlier, a common technique for noise spectrum estimation is to apply temporal recursive smoothing during the frames when only noise is present. Now, we have the following hypotheses:

$$\begin{aligned} H_0 &: \lambda_{n,k}(l+1) = a_n \lambda_{n,k}(l) + (1-a_n)|X_k(l)|^2, \\ H_1 &: \lambda_{n,k}(l+1) = \lambda_{n,k}(l), \end{aligned} \tag{10}$$

where $0 < a_n < 1$ is a smoothing parameter.

Let $p_{s,k}(l) = p(H_1|X_k(l))$ denote the conditional speech presence probability at time frame l. Hence, we can write (10) as follows:

$$\begin{aligned} \lambda_{n,k}(l+1) &= \lambda_{n,k}(l) p_{s,k}(l) \\ &\quad + \left[a_n \lambda_{n,k}(l) + (1-a_n)|X_k(l)|^2\right](1 - p_{s,k}(l)) \\ &= \tilde{a}_{n,k}(l)\lambda_{n,k}(l) + (1 - \tilde{a}_{n,k}(l))|X_k(l)|^2, \end{aligned} \tag{11}$$

where

$$\tilde{a}_{n,k}(l) = a_n + (1-a_n) p_{s,k}(l) \tag{12}$$

is a time-varying smoothing parameter. We can see that the noise spectrum is estimated by averaging past power spectral values, using a smoothing parameter that is adjusted by the speech presence probability $p_{s,k}(l)$. In order to determine $p_{s,k}(l)$, speech absence is calculated by looking at the ratio of the local energy of the noisy signal and its minimum within a certain time frame. For details on the estimation of $p_{s,k}(l)$ please confer [26].

3.2. Decision directed a priori SNR estimation

The decision directed (DD) estimation approach for the estimation of ξ_k, the *a priori* SNR, was proposed in [28]. Firstly, the Wiener gain is introduced as the following ratio:

$$\zeta_k = \frac{\xi_k}{\xi_k + 1}. \tag{13}$$

Now, we can define the estimator for ξ_k:

$$\xi_k(l) = \alpha_a \zeta_k^2(l-1)\gamma_k(l-1) + (1-\alpha_a)\max\{\gamma_k(l) - 1, 0\}, \tag{14}$$

where $0 < \alpha_a < 1$ is a smoothing parameter.

The noise spectrum $\lambda_{n,k}$ and the *a priori* SNR ξ_k are continuously updated via the MCRA and DD methods, respectively, and are afterwards used in the RRD VAD. An overview on advancements in speech enhancement can be found in [29].

4. Speech corpus and metrics for voice activity detection evaluation

In order to analyze the supervised learning based VAD algorithms and performance thereof, we used the NOIZEUS speech corpus by [30]. Although the corpus was originally created

for testing speech enhancement algorithms, we used it for the following reasons: (i) the recordings are of high quality and were made in a sound-proof booth, (ii) it offers eight different types of noises from AURORA database by [31] which corrupt the original recordings at four different SNR levels, (iii) the recordings were made by six different speakers—three male and three female, (iv) it uses the IEEE sentence database which contains phonetically-balanced sentences with relatively low word-context predictability, and (v) the corpus is available to researchers free of charge. The percentage of the speech segments amounted to 61.28%, which is as twice as high as compared to [1], and [4], but less than 5% higher than in the cases of [5] and [8]. The recordings were sampled at the rate of 25 kHz and were later downsampled to 8 kHz. The total length of all the recordings was 80.04 s, which offered, with 50% overlap and frame length of $L = 256$, in total 5000 frames for detection. However, in order to test the performance and train the classifier for different types of noises and noise levels, we have added to the clean speech also versions corrupted with babble (SNR 15 dB, 10 dB, 5 dB), car (SNR 15 dB, 10 dB, 5 dB) and white Gaussian noise (SNR 20 dB, 15 dB, 10 dB). In total, this gave us 50000 frames for evaluation.

Usually, in order to test and train the algorithms, the speech segments are hand-labeled. However, in the present work we used signal energy calculated via Parseval's theorem as the indicator of speech presence, which enabled automatic frame labeling. We find this approach justifiable in the case of the NOIZEUS corpus, since the clean recordings were made in a sound-proof booth resulting with the speech-absent frames having energy a thousand times lower than the weakest speech frame.

The evaluation metrics we used are based on the standard elements of the confusion matrix: true positive (TP)—voice classified as voice, true negative (TN)—silence classified as silence, false positive (FP)—silence classified as voice, false negative (FN)—voice classified as silence. We also used speech detection rate (SDR)—percentage of speech frames classified as speech, and false alarm rate (FAR)—percentage of noise frames classified as speech. The former and latter are calculated as follows:

$$\mathrm{SDR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad \mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{15}$$

These two rates are actually used in order to draw a receiver operating characteristics (ROC) curve. An ROC curve is a two-dimensional depiction of classifier performance. Usually, they are produced by graphing pairs of SDR and FAR values as a function of changes in the threshold value. To compare different classifiers it is practical to reduce the information in the ROC curve to a single scalar value. A common method is to evaluate the area under an ROC curve (AUC). For an example, since both the SDR and FAR take values in the range of $[0, \ldots, 1]$, for a perfect classifier the AUC value would be 1, since it is able to make a perfect SDR without any false alarms. A completely random classifier would have AUC value of 0.5, since the ROC curve would be a diagonal line in the SDR–FAR space. This would be equivalent to predicting based on fair coin tosses. More on the ROC curves and metrics for evaluation of classifiers can be found in [32, 33].

Another balanced measure of classification performance with respect to all elements is the Matthews correlation coefficient (MCC) which we chose as additional metric for performance comparison. It is calculated as follows [32]:

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FN})(\mathrm{TP} + \mathrm{FP})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}. \tag{16}$$

The MCC is always between -1 and $+1$, where -1 indicates total disagreement and $+1$ indicates total agreement. The MCC is 0 for completely random predictions. If two variables are independent, then their MCC is 0. The converse in general is not true.

5. Input variable selection based on partial mutual information

Before we start with classification, we need to choose input variables, i.e. features, upon which the classifiers will make decision and which, in effect, will be combined to form a strong classifier. We already mentioned that LR is one of the features, but we hypothesize that by adding other features we could improve the clasiffication results.

5.1. Partial Mutual Information

The partial mutual information (PMI) based input variable selection (IVS) algorithm used in [34, 35] overcomes two main issues that limit the applicability of many IVS techniques. Those are the underlying assumption of linearity and redundancy within the available data. The way that PMI IVS works is that it first selects the most informative input variable, then it searches for the next most informative variable but by taking into account information already received from the previously selected variable. This process continues until an introduction of an additional input variable increases the mean squared error of the prediction, i.e. the square of the expected value minus the label, or PMI drops below a certain threshold. Hereafter, we present the mathematical background of the PMI IVS.

Assuming y is a classification outcome, i.e. signal frame label, x is a currently considered input variable (feature), and $\mathbf{z}$ is a set of previously selected variables, partial mutual information in x about y given $\mathbf{z}$ is formulated as follows:

$$\mathrm{PMI} = \iint p_{u,v}(u, v) \ln \frac{p_{u,v}(u, v)}{p_u(u) p_v(v)} \mathrm{d}u\mathrm{d}v \,, \tag{17}$$

where $u = y - \mathrm{E}[y|\mathbf{z}]$, $v = x - \mathrm{E}[x|\mathbf{z}]$, and $\mathrm{E}[\,.\,]$ is the expectation operator.

In order to obtain probability density functions for PMI from the data, we used kernel density estimators (KDEs). E.g., in order to calculate $\mathrm{E}[x|\mathbf{z}]$ we used the following KDE:

$$\hat{p}(x, \mathbf{z}) = \frac{1}{n} \frac{1}{\left(\sqrt{2\pi}h\right)^d \sqrt{|\Sigma|}} \sum_{i=1}^{n} \exp - \frac{\|[x\ \mathbf{z}]^T - [x_i\ \mathbf{z}_i]^T\|_\Sigma}{2h^2}, \tag{18}$$

where $\|[x\,\mathbf{z}]^T - [x_i\,\mathbf{z}_i]^T\|_\Sigma = ([x\,\mathbf{z}] - [x_i\,\mathbf{z}_i])\,\Sigma^{-1}\left([x\,\mathbf{z}]^T - [x_i\,\mathbf{z}_i]^T\right)$ is the Mahalanobis distance, and h is the kernel bandwidth, for

which we used the Gaussian reference bandwidth throughout this paper:

$$h = \left(\frac{4}{d+2}\right)^{\frac{1}{d+4}} n^{-\frac{1}{d+4}} , \tag{19}$$

where d is the dimension of the multivariate variable set, and n is the sample size.

Note that for $\mathrm{E}[x|\mathbf{z}]$ we need $\hat{p}(x|\mathbf{z})$. If we take

$$\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}, \tag{20}$$

we get

$$\hat{p}(x|\mathbf{z}) = \frac{1}{n} \frac{1}{\left(\sqrt{2\pi}h\right)^d \sqrt{|\bar{\Sigma}|}} \sum_{i=1}^{n} \exp - \frac{\|x^T - \bar{x}_i^T\|_{\bar{\Sigma}}}{2h^2} , \tag{21}$$

where $\bar{\Sigma} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}$ and $\bar{x}_i = x_i + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \mathbf{z}_i)$.

Finally,

$$\mathrm{E}[x|\mathbf{z}] = \sum_{i=1}^{n} w_i \left[x_i + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \mathbf{z}_i)\right] , \tag{22}$$

where each sample is weighted by its weighting factor introduced in [34]:

$$w_i = \frac{\exp\left(-\frac{\|\mathbf{z}^T - \mathbf{z}_i^T\|_{\Sigma_{zz}}}{2h^2}\right)}{\sum_{j=1}^{n} \exp\left(-\frac{\|\mathbf{z}^T - \mathbf{z}_j^T\|_{\Sigma_{zz}}}{2h^2}\right)} . \tag{23}$$

The pseudocode of IVS based on PMI utilized in the present paper is given in Algorithm 1.

Algorithm 1: Input variable selection based on partial mutual information.

Input: sets of considered variables $X = \{x_1, x_2, \dots\}$ and labels $Y = \{y_1, y_2, \dots\}$
Output: set of chosen input variables $Z = \{z_1, z_2, \dots\}$
Initialize $Z \leftarrow \emptyset$
Initialize $u_{MSE} \leftarrow \infty$
while $X \neq \emptyset$ **do**
  Construct an estimator $\mathrm{E}[y|\mathbf{z}]$
  Calculate $u \leftarrow y - \mathrm{E}[y|\mathbf{z}]$
  if $u_{MSE} < \mathtt{mse}(u)$ **then**
    Remove previously added x from Z
    exit
  $u_{MSE} \leftarrow \mathtt{mse}(u)$
  foreach $x \in X$ **do**
    Construct an estimator $\mathrm{E}[x|\mathbf{z}]$
    Calculate $v \leftarrow x - \mathrm{E}[x|\mathbf{z}]$
    Determine the PMI $I(v, u)$
  Determine $x = x_s$ (i.e. v) which maximizes $I(v, u)$
  if $I(v, u) < I_{min}$ **then**
    exit
  Add x_s to Z

5.2. Input variable set

In the ensuing paragraphs we present the features that form the potential input variable set. Each of them was analyzed as a standalone detector and as a candidate for the reduced input vector by the PMI IVS.

Magnitude of the DFT coefficients. A K-point transform was used to analyze the spectrum of the recorded frames. The magnitude of the first 32 coefficients of the transform were used as a feature for the classifier.

Zero-crossing rate. The Zero Crossing Rate (ZCR) of a signal is the rate of sign changes along the signal. It is defined as follows:

$$f_{ZCR} = \sum_{i=2}^{L} Z_i,$$
$$\text{where } Z_i = \begin{cases} 1, & \text{if } \operatorname{sign}\{x(i)\} - \operatorname{sign}\{x(i-1)\} \neq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

Human voice consists of voiced and unvoiced sounds. Voiced sounds have higher ZCR value than the unvoiced sounds do. Therefore, it is a reasonable assumption that ZCR of either voiced or unvoiced parts of speech will be different than the ZCR of noise in the silent periods.

Spectral flux. Spectral flux (SF) measures how quickly the spectrum of the signal is changing. It is calculated by comparing the power spectrum of the current frame with the power spectrum of the previous frame.

$$f_{SF} = \left| \sum_{k=1}^{K} \left(|X_k(l)|^2 - |X_k(l-1)|^2 \right) \right| \tag{25}$$

Speech changes quickly between voiced and unvoiced parts, thus resulting with high SF values.

Spectral rolloff. Spectral rolloff (SR) is defined as the a-quantile of the total energy in $|X_k|^2$. It is a frequency under which a fraction of the total energy is found. If K is the length of the signal DFT, then SR can be defined as:

$$f_{SR} = \max_{y} \left\{ y : a > \frac{\sum_{k=1}^{y} |X_k|^2}{\sum_{k=1}^{K} |X_k|^2} \right\} \tag{26}$$

Spectral rolloff was calculated at six quantiles equally spaced in $[0, 1]$.

Mel-frequency cepstral coefficients. Mel-frequency analysis is a technique inspired by human sound perception. The human ear acts as a filter and concentrates only on specific spectral components. The filters are non-uniformly spaced on a frequency scale, and their density is higher in the low frequency regions. The MFCCs are calculated in several steps: (i) the magnitude spectrum $|X_k|$ is filtered with a bank of non-uniformly spaced overlapping triangular filters, (ii) the logarithm is taken, and (iii) the MFCC are obtained by computing the discrete cosine transform of the result. In [36] where authors consider a voice conversion system, MFCC feature is identified as a feature that does not consider any particular speech model, i.e. feature that is useful for general voice activity detection, without considering any speaker in particular.

Power-normalized cepstral coefficients. In [37, 38] a feature extraction algorithm called power-normalized cepstral coefficients (PNCC) was proposed, which instead of log nonlinearity like MFCC uses power-law nonlinearity and a gammatone filterbank. In [37] it was shown to outperform MFCC, among others, in speech recognition accuracy. After adapting the algorithm proposed in [37] to our scenario, we have used the first thirteen PNCCs which were the result of a 20 element gammatone prefiltering.

Spectral centroid. Spectral centroid (SC) is a statistic that measures where most of the power of a speech segment is spectrally located. It is defined as follows:

$$f_{SC} = \frac{\sum_{k=1}^{K} k|X_k|^2}{\sum_{k=1}^{K} |X_k|^2}. \tag{27}$$

Spectral bandwidth. Spectral bandwidth (SBW) describes spreading of the spectral components with respect to the spectral centroid:

$$f_{SBW} = \sqrt{\frac{\sum_{k=1}^{K} (k - f_{SC})^2 |X_k|^2}{\sum_{k=1}^{K} |X_k|^2}}. \tag{28}$$

Feature aggregation. In total the following features were aggregated: 1 LR, 32 DFT magnitude coefficients, 1 ZCR, 1 SF, 6 SR quantiles, 15 mel-frequency cepstral coefficients, 13 power normalized cepstral coefficients, 1 SC and 1 SBW. Thus, we had a feature vector of 71 for input variable analysis. Similar approach was used in [39, 12] for music classification.

5.3. Individual feature performance and IVS results

Each of the afore presented features can be considered as a detector in itself, whose performance might indicate the suitability of being an element in the input vector. As an intuitive preliminary analysis, we utilized the ROC curves, i.e. the related AUC score, of each feature evaluated on the whole data set at once. Table 1 shows the AUC for all the features presented in the current section. We can see that the LR has the highest score, followed by the first PNCC, SF, the first MFCC coefficient, while the third and ninth PNCC have the lowest score. Furthermore, ROC curves for five features with the highest AUC score are depicted in Fig. 1, while the values of three features with the highest AUC score along with the label for 200 frames are depicted in Fig. 2.

Due to high memory requirements the analysis based on partial mutual information was carried out on the set consisting of the clean signal, and its versions corrupted with babble (SNR 10 dB), car (SNR 10 dB), and white Gaussian noise (SNR 15 dB) separately. The analysis on each set was stopped once the addition of another feature caused increase in the mean squared error. Based on the results we kept those features that were chosen in at least two sets: the LR, DFT indexes 7, 8, 9, 11, the 1st and 2nd SR, the 1st MFCC, SC, SBW, and 1st, 2nd and 3rd PNCC. It is interesting to note that the PMI algorithm chose the 3rd PNCC as a good feature, although it has by far the lowest AUC score than many other features. However, the PMI chooses features which bring additional information when all

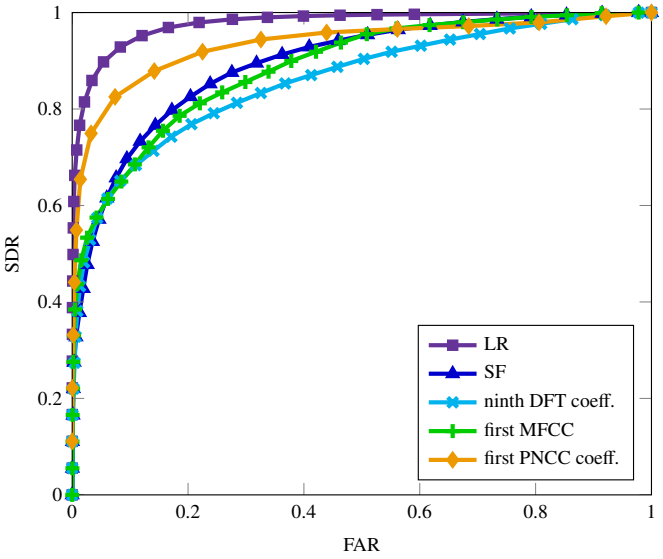

Figure 1: ROC curves for the five features with the highest AUC score.

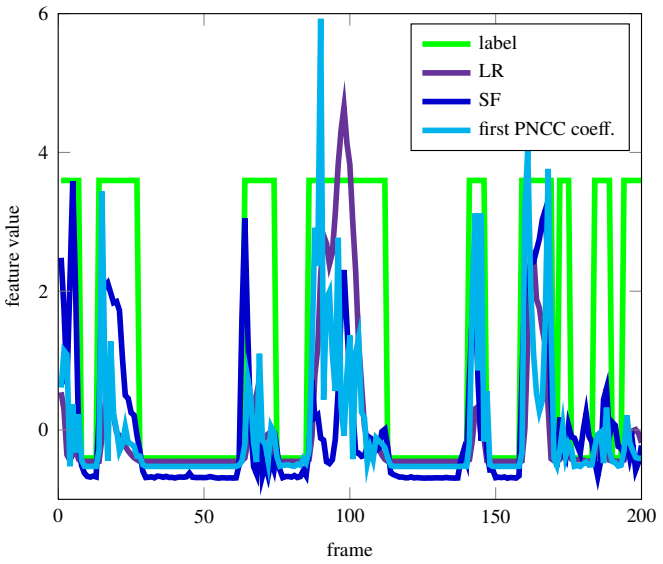

Figure 2: Feature values for a random segment of 200 frames corrupted with babble noise (15 dB SNR).

the information from other features is taken into account, meaning that in certain scenarios the 3rd PNCC contributed to correct classification. In total this amounts to 13 features forming a reduced vector of input variables, which is an 82% decrease in the size of the feature vector.

Although from Fig. 1 we can see that the LR as a standalone detector outperforms other features, we conjecture and shall test (i) that a trained classifier based on LR and other features should outperform a statistical model-based detector based on LR, and (ii) that a detector with carefully chosen reduced input vector should not significantly underperform the detector based on a full feature vector. We shall test these hypotheses on 50000 learning examples and by meticulous analysis with ROC curves and the AUC metric.

Table 1: AUC score of all the features.

	Feature	AUC		Feature	AUC		Feature	AUC
1.	LR	0.978	25.	31st DFT	0.708	49.	3rd DFT	0.632
2.	1st PNCC	0.936	26.	4thSR	0.708	50.	1stSR	0.624
3.	SF	0.895	27.	22nd DFT	0.708	51.	12th MFCC	0.622
4.	1st MFCC	0.888	28.	32nd DFT	0.706	52.	4th DFT	0.619
5.	9th DFT	0.861	29.	23rd DFT	0.706	53.	4th MFCC	0.609
6.	15th DFT	0.815	30.	21st DFT	0.704	54.	14th MFCC	0.609
7.	8th DFT	0.810	31.	20th DFT	0.702	55.	4th PNCC	0.603
8.	6th MFCC	0.809	32.	19th DFT	0.702	56.	6th DFT	0.602
9.	16th DFT	0.805	33.	30th DFT	0.700	57.	9th MFCC	0.601
10.	14th DFT	0.793	34.	24th DFT	0.694	58.	7th PNCC	0.597
11.	10th DFT	0.786	35.	2nd MFCC	0.692	59.	3rd MFCC	0.586
12.	17th DFT	0.767	36.	5th MFCC	0.686	60.	8th MFCC	0.583
13.	13th DFT	0.765	37.	29th DFT	0.680	61.	13th MFCC	0.566
14.	12th DFT	0.751	38.	25th DFT	0.663	62.	10th PNCC	0.564
15.	11th DFT	0.747	39.	1st DFT	0.661	63.	11th PNCC	0.561
16.	7th MFCC	0.743	40.	28th DFT	0.660	64.	13th PNCC	0.554
17.	3rd SR	0.739	41.	2nd PNCC	0.658	65.	15th MFCC	0.548
18.	ZCR	0.731	42.	6th PNCC	0.655	66.	5th PNCC	0.545
19.	18th DFT	0.726	43.	2nd DFT	0.655	67.	10th MFCC	0.541
20.	2nd SR	0.725	44.	7th DFT	0.652	68.	8th PNCC	0.519
21.	SBW	0.722	45.	27th DFT	0.648	69.	12th PNCC	0.518
22.	5thSR	0.720	46.	11th MFCC	0.647	70.	3rd PNCC	0.511
23.	6thSR	0.719	47.	26th DFT	0.644	71.	9th PNCC	0.505
24.	SC	0.713	48.	5th DFT	0.637			

6. Quantitative evaluation of SVM, Boost, and ANN based voice activity detectors

In the present paper we utilized and compared three supervised learning algorithms; SVM, Boost, and ANN, which were to classify if a signal frame contains speech or not based on the full and the reduced feature set generated by algorithm in Section 5. The three have different approaches to learning and all have their advantages, and we shall briefly introduce each in the following paragraphs. But it is important to notice at this point that the goal of the present paper is not to provide a detailed tutorial in either of the classifiers, but to analyze and compare the performance of the three for the specific purpose of voice activity detection based on various features and not in general. For training and testing the three learning algorithms we used the OpenCV library [40].

Essentially, SVM [41, 42] is a learning algorithm that constructs a hyperplane or a set of hyperplanes which define boundaries for the data to be discriminated. The data, most often, is not linearly separable and this problem is addressed by SVM in a way that non-linearly maps the input vector with a kernel function to a high-dimensional feature space. They can also be used in regression tasks, but in the present paper we use them in the context of a binary classifier. An introduction to the theory behind SVM and some practical insights can be found in [43]. In the present paper we used C-support vector classification and radial basis function RBF as the kernel function.

The main idea behind boosting algorithms is to use many simple detectors which should have performance a bit better than 50% at least (i.e. better than random guessing)—these are called weak classifiers—and combine them to obtain highly accurate classifier—usually called a strong classifier. In its original form, Boost handles binary classification problems only, although there are extensions to handle multi-class and even multi-label classification problems [44]. In the present paper, a variant of the Boost algorithm proposed in [45] called Real Boost is used [46].

The ANNs are a product of the desire to imitate the workings of the biological brain. They involve a network of simple processing elements (artificial neurons) which can exhibit complex global behavior. One of the most important properties of ANNs is the ability to approximate any continuous function up to a given precision. They have been extensively used in both classification and regression tasks and more on the ANNs can be found in [47]. In the present paper we utilize a static multilayer perceptron network (MLP) with a sigmoid activation function, a single hidden layer with 5 neurons, while the network parameters are learned using the resilient propagation (RPROP) algorithm [48].

6.1. Evaluation of the supervised learning VAD algorithms

In this section we analyze the performance of the classifiers. The data was constructed by concatenating the clean signal with its corrupted versions thus, with frame length of $L = 256$ samples, yielding 50000 examples for evaluation. For the full input vector we had 71 features, while the reduced input vector

consisted of 13 features. Prior to the learning process, all the features were scaled in a way to have a zero mean value and standard deviation of one.

The evaluation was performed by K-fold cross-validation. Essentially, the original dataset was partitioned randomly into K subsets of equal size. Of the K subsets, one was retained for testing the classifier while the other $K-1$ subsets were used for training. The cross-validation process was repeated K times thus yielding K results which were used for drawing the average ROC curves. As discussed in [33], by drawing just an ROC curve of different classifiers and seeing which one dominates to assess the performance might be misleading, since we do not have a measure of variance. Therefore, it is suggested to generate results from several test subsets, by a cross-validation or bootstrap method, and average these results in order to obtain a measure of variance. The ROC curves can be either averaged vertically by fixing FAR and averaging over SDR, or by the threshold, where for each threshold value an SDR–FAR pair is found and their values are averaged thus yielding both vertical and horizontal variance. In the present paper we used 10-fold cross-validation and threshold averaging for evaluation of the VAD algorithms.

Firstly, we compared intra-classifier performance, i.e. performance of each classifier working with either the full or the reduced input vector. Henceforth, all the figures depicting ROC curves have for each point a confidence interval of three standard deviations included, along with the AUC score and three standard deviations thereof. These deviations indicate just how consistent the classifier performance was with respect to different cross-validation sets. Figure 3 shows the averaged ROC curves and their AUC score for the SVM, from which we can see that the classifier with the reduced feature set did not significantly underperform compared to the classifier trained on the full feature set. In Fig. 4 we can see a bit different result for the Boost classifier. In this case the classifier showed practically equal performance both in the mean and standard deviation when being trained on the full and the reduced input set. Finally, Fig. 5 shows the averaged ROC curves and their AUC score for the ANN. It performed slightly better in the mean and standard deviation of the AUC score with the full input vector, but overall exhibited larger deviations than any of the other two classifiers. This means that it did not perform as consistently over all the subsets.

To conclude the intra-classifier analysis, we can assert that the results supported our second hypothesis from the Section 5: neither of the classifiers significantly underperformed when being trained on the reduced input vector formed by a careful IVS. Henceforth, we shall only include in the analysis the classifiers trained on the reduced input vector.

For the inter-classifier performance we also included the statistical model-based detector presented in Section 2.2 which too was evaluated by K-fold cross-validation. Since it does not require training it was simply tested on the same K subsets and these results were averaged. Figure 6 shows ROC curves for the three supervised learning classifiers and the RRD detector based on LR, from which we can see that the supervised learning approach with several additional features can significantly

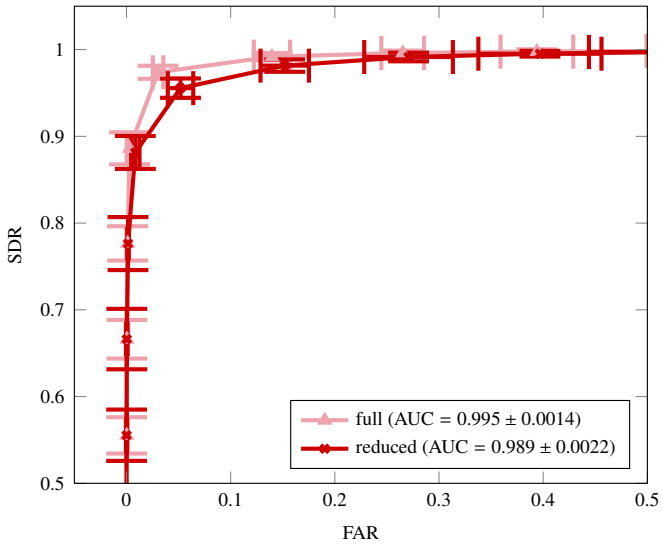

Figure 3: Averaged ROC curves for the SVM classifier with the full and reduced input vector.

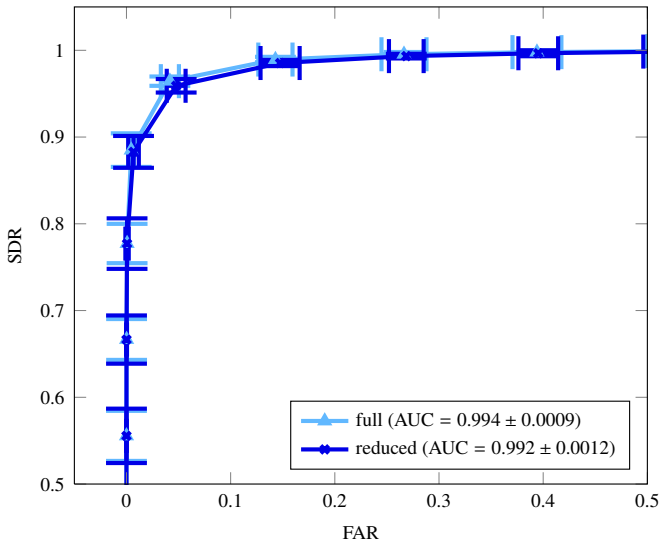

Figure 4: Averaged ROC curves for the Boost classifier with the full and reduced input vector.

increase the performance of a detector. Moreover, judging from the AUC scores shown in Fig. 6 we can assert that the Boost classifier slightly outperforms the other classifiers, since it has the largest AUC mean value and the smallest AUC standard deviation. Furthermore, by inspecting Figures 3, 4, and 5 we can also see that Boost overall exhibited smaller deviations in the ROC curves, which further tips the balance in Boost's favor.

During the K-fold cross-validation we also monitored the performance of the trained classifiers for each subset by calculating the SDR, FAR, and MCC presented in Section 4. Since all the classifiers were trained to output a value between -1, for non-speech, and 1, for speech frames, we set the threshold to zero, thus all the frames with score larger or equal to zero were classified as containing speech, while the other were classified as non-speech frames. This essentially would correspond

Table 2: Averaged statistical scores of the trained classifier performance.

		SDR [%]	FAR [%]	ERR [%]	MCC±3σ_{MCC}
SVM	full	96.73	2.26	5.53	0.944 ± 0.0141
	red	94.47	3.71	9.24	0.906 ± 0.0183
Boost	full	95.79	3.35	7.56	0.923 ± 0.0132
	red	95.10	3.75	8.65	0.912 ± 0.0150
ANN	full	95.23	3.90	8.67	0.912 ± 0.0189
	red	93.43	5.05	11.62	0.882 ± 0.0309

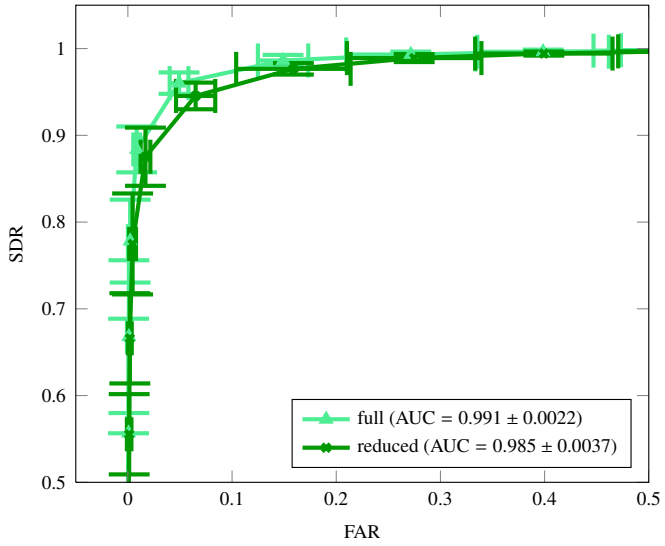

Figure 5: Averaged ROC curves for the ANN classifier with the full and reduced input vector.

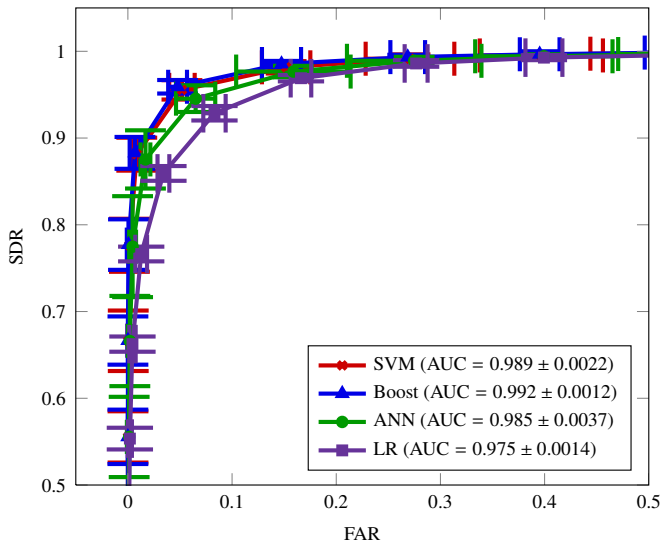

Figure 6: Averaged ROC curves for all the classifiers with the reduced input vector and the detector based solely on the LR.

to only a single point in the ROC curve graph, but it is very practical since it provides a tangible sense of performance for a single threshold value. The average of these statistical scores for the aforementioned 10 subsets is shown in Table 2, where we also provide error rate (ERR = (100 - SDR) + FAR) since it is often used in other works.

To conclude the inter-classifier performance, from the above presented results we can see that the classifiers significantly outperformed the statistical model-based detector, and that due to having the highest AUC score with the smallest standard deviation, and exhibiting no significant deviations anywhere in the ROC curve, the Boost algorithm had the advantage over the other algorithms for this specific application of speech activity detection based on various features. Therefore, we can assert that the results supported our first hypothesis from Section 5 that a trained classifier based on LR and other features should outperform a statistical model-based detector based on LR.

These experiments were designed so as to find a LR model that will show the best results [23], which we would then extend with features meticulously analyzed with PMI IVS and encompass it all in a supervised learning framework which showed the best and most consistent performance. Furthermore, the corpus that we used is freely available to all researchers [30] which will enable direct comparison of detection algorithms in the future. Comparison of our results to works which utilized a supervised learning approach [15–17, 20, 18, 19] is not straightforward due to utilization of a different speech corpus, graphical result representation (no score presented) or non-direct metric (word accuracy rate in speech recognition). However, some do provide ERR score for different noise levels and types which we will use for crude comparison with our results. For an example, in [15] the best ERR was 5.38 % and 13.47 % for vehicle and office noise, respectively, while [16] reports 9.4 % and 20.9 % for vehicle and babble noise, respectively. In [20] authors report ERR from 7.83 % to 41.39 % for different test sequences. The authors in [17] report a score named equal error rate for which equality 1-SDR=FAR holds. For three different datasets they report equal error rate of 8.0 %, 13.1 %, and 19.0 % for an SVM trained on MFCC. Comparing these results with Table 2 we can see that our results do not deviate and are in the rank of their performance. However, since different datasets were used in these papers, a direct comparison is not possible.

7. Conclusion

In the present paper, we have presented the theory behind statistical model-based VADs and derived the LR for Rayleigh-Rice distribution based VAD. Furthermore, we have introduced in total 70 additional features which were combined with the RRD based VAD to form an input vector for the supervised learning classifiers. The input vector was extensively analyzed by a partial mutual information algorithm in order to single out the most informative features and by AUC score analysis to test the capability of each feature to serve as a VAD . The results yielded a 13 element reduced input vector. We have focused on SVM, Boost and ANN classifiers, whose performances were mutually compared both with the full and the reduced input vector. The algorithms were tested on the NOIZEUS speech corpus consisting of recordings made by six different speakers and which were corrupted by three different types and levels of noises. The performance evaluation was based on a 10-fold cross-validation and compared on threshold averaged ROC curves, AUC score and MCC. Firstly, the results showed that the performance was not undermined by utilizing the vector with the reduced number of features. Secondly, although the statistical model-based VAD by itself is a much better detector than any of the other utilized features, a combination of the latter and the former in the form of a trained classifier produced a VAD with significantly better performance. Finally, inter-classifier analysis showed similar performance of the three, with a slight advantage in the direction of the Boost classifier, since it had the highest AUC score and the smallest variability in the threshold averaged ROC curves, indicating a consistent performance over all the test subsets.

The presented approach consisting of aggregating various features, performing input variable selection by a partial mutual information algorithm whereat a reduced input vector is created, and training a classifier for voice activity detection, is quite generic. It can be used on any combination of features and, indeed, is not limited just to voice activity detection. In order to further increase the VAD performance or tailor it to specific scenarios, a cascaded classifier architectures can be utilized, for which the presented approach would be indivertible.

ACKNOWLEDGMENTS

The authors would like to thank Vlaho Petrović of the University of Zagreb, Faculty of Electrical Engineering and Computing for valuable discussions concerning the partial mutual information method, and the reviewers for insightful comments that helped to improve this manuscript and for drawing our attention to the power-normalized cepstral coefficients as a potential feature in the input vector.

References

[1] J. Sohn, N. S. Kim, W. Sung, A Statistical Model-Based Voice Activity Detection, IEEE Signal Processing Letters 6 (1) (1999) 1–3.

[2] Y. D. Cho, K. Al-Naimi, A. Kondoz, Improved Voice Activity Detection Based on a Smoothed Statistical Likelihood Ratio, in: Proceeding of the International Conference on Acoustics, Speech and Signal Processing, 2001, pp. 737–740.

[3] D. K. Kim, J.-H. Chang, A Subspace Approach Based on Embedded Prewhitening for Voice Activity Detection., The Journal of the Acoustical Society of America 130 (5) (2011) EL304–10.

[4] J.-H. Chang, N. S. Kim, Voice Activity detection Based on Complex Laplacian Model, Electronics Letters 39 (7) (2003) 632.

[5] J.-H. Chang, J. W. Shin, N. S. Kim, Voice Activity Detector Employing Generalised Gaussian Distribution, Electronics Letters 40 (24) (2004) 25–26.

[6] J. Ramírez, J. C. Segura, J. M. Górriz, L. García, Improved Voice Activity Detection Using Contextual Multiple Hypothesis Testing for Robust Speech Recognition, IEEE Transactions on Audio Speech and Language Processing 15 (8) (2007) 2177–2189.

[7] J. Ramírez, J. M. Górriz, J. C. Segura, Statistical Voice Activity Detection Based on Integrated Bispectrum Likelihood Ratio Tests, Journal of the Acoustical Society of America 121 (5) (2007) 2946–2958.

[8] J. M. Górriz, J. Ramírez, E. W. Lang, C. G. Puntonet, I. Turias, Improved Likelihood Ratio Test Based Voice Activity Detector Applied to Speech Recognition, Speech Communication 52 (2010) 664–677.

[9] K.-H. Woo, T.-Y. Yang, K.-Y. Park, C. Lee, Robust Voice Activity Detection Algorithm for Estimating Noise Spectrum, Electronics Letters 36 (2) (2000) 180–181.

[10] Q. Li, J. Zheng, A. Tsai, Q. Zhou, Robust Endpoint Detection and Energy Normalization for Real-Time Speech and Speaker Recognition, IEEE Transactions on Speech and Audio Processing 10 (3) (2002) 146–157.

[11] M. Marzinzik, B. Kollmeier, Speech Pause Detection for Noise Spectrum Estimation by Tracking Power Envelope Dynamics, IEEE Transactions on Speech and Audio Processing 10 (6) (2002) 341–351.

[12] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, B. Kégl, Aggregate Features and ADABOOST for Music Classification, Machine Learning 65 (2-3) (2006) 473–484.

[13] P. Dhanalakshmi, S. Palanivel, V. Ramalingam, Classification of Audio Signals using AANN and GMM, Applied Soft Computing 11 (1) (2011) 716–723.

[14] F. F. Li, T. J. Cox, A Neural Network Model for Speech Intelligibility Quantification, Applied Soft Computing 7 (1) (2007) 145–155.

[15] J. W. Shin, J.-H. Chang, N. S. Kim, Voice Activity Detection Based on Statistical Models and Machine Learning Approaches, Computer Speech & Language 24 (3) (2010) 515–530.

[16] Q.-H. Jo, J.-H. Chang, J. Shin, N. Kim, Statistical Model-Based Voice Activity Detection Using Support Vector Machine, IET Signal Processing 3 (3) (2009) 205.

[17] T. Kinnunen, E. Chernenko, M. Tuononen, P. Fränti, H. Li, Voice Activity Detection Using MFCC Features and Support Vector Machine, in: Int. Conf. on Speech and Computer (SPECOM07), 2007, pp. 556–561.

[18] J. Ramirez, P. Yélamos, J. M. Górriz, J. C. Segura, SVM-Based Speech Endpoint Detection Using Contextual Speech Features, Electronics Letters 42 (7) (2006) 426–428.

[19] J. Ramírez, P. Yélamos, J. M. Górriz, J. Segura, L. García, Speech / Non-Speech Discrimination Combining Advanced Feature Extraction and SVM Learning, in: International Conference on Spoken Language Processing (INTERSPEECH 2006), 2006, pp. 1662–1665.

[20] D. Enqing, L. Guizhong, Z. Yatong, Z. Xiaodi, Applying Support Vector Machines to Voice Activity Detection, in: 6th International Conference on Signal Processing, 2002, pp. 1124 – 1127.

[21] ITU-T, A Silence Compression Scheme for G.729 Optimized for Terminals Conforming to Recommendation v70. ITU-T Rec. G. 729, Annex B, Tech. rep. (1996).

[22] X.-L. Zhang, J. Wu, Linearithmic Time Sparse and Convex Maximum Margin Clustering., IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics (2012) 1–24.

[23] I. Marković, H. Domitrović, I. Petrović, Comparison of Statistical Model-Based Voice Activity Detector for Mobile Robot Speech Applications, 10th IFAC Symposioum on Robotic Control 2012 (SYROCO2012).

[24] E. Mumolo, M. Nolich, G. Verchelli, Algorithms for Acoustic Localization Based on Microphone Array in Service Robotics, Robotics and Autonomous Systems 42 (2) (2003) 69–88.

[25] R. McAulay, M. Malpass, Speech Enhancement Using a Soft-Decision

Noise Suppression Filter, IEEE Transactions on Acoustics Speech and Signal Processing 28 (1980) 137–145.
[26] I. Cohen, B. Berdugo, Speech Enhancement for Non-Stationary Noise Environments, Signal Processing 81 (2001) 283–288.
[27] I. Cohen, Noise Spectrum Estimation in Adverse Environments: Improved Minima Controlled Recursive Averaging, Speech and Audio Processing 11 (2003) 466–475.
[28] Y. Ephraim, D. Malah, Speech Enhancement Using a Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator, Speech and Signal Processing (1984) 1109–1121.
[29] Y. Ephraim, I. Cohen, Recent Advancements in Speech Enhancement, in: C. Dorf (Ed.), Circuits, Signals, and Speech and Image Processing, Taylor and Francis, 2006.
[30] Y. Hu, P. C. Loizou, Subjective Comparison and Evaluation of Speech Enhancement Algorithms., Speech Communication 49 (7) (2007) 588–601.
[31] D. Pearce, H.-G. Hirsch, The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions, in: ISCA ITRW ASR2000, 2000, pp. 29–32.
[32] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, H. Nielsen, Assessing the Accuracy of Prediction Algorithms for Classification: an Overview, Bioinformatics Review 16 (5) (2000) 412–424.
[33] T. Fawcett, ROC Graphs: Notes and Practical Considerations for Researchers, Tech. rep., HP Labs Tech Report (2004).
[34] A. Sharma, Seasonal to Interannual Rainfall Probabilistic Forecasts for Improved Water Supply Management: Part 1 A Strategy for System Predictor Identification, Journal of Hydrology 239 (2000) 232–239.
[35] R. J. May, H. R. Maier, G. C. Dandy, T. G. Fernando, Non-Linear Variable Selection for Artificial Neural Networks using Partial Mutual Information, Environmental Modelling & Software 23 (10-11) (2008) 1312–1326.
[36] R. Laskar, D. Chakrabarty, F. Talukdar, K. S. Rao, K. Banerjee, Comparing ANN and GMM in a Voice Conversion Framework, Applied Soft Computing 12 (11) (2012) 3332–3342.
[37] C. Kim, R. M. Stern, Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition, IEEE Transactions on Audio Speech and Language Processing in print.
[38] C. Kim, R. M. Stern, Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recoginition, in: International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012), 2012, pp. 4101–4104.
[39] T. Diethe, J. Shawe-Taylor, Linear Programming Boosting for the Classification of Musical Genre, in: Neural Information Processing Systems (NIPS), Whistler, Canada, 2007.
[40] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools.
[41] B. E. Boser, I. M. Guyon, V. N. Vapnik, A Training Algorithm for Optimal Margin Classifers, in: 5th Annual ACM Workshop on COL, 1992, pp. 144–152.
[42] C. Cortes, V. Vapnik, Support-Vector Networks, Machine Learning 20 (1995) 273–297.
[43] C.-C. Chang, C.-J. Lin, LIBSVM : A Library for Support Vector Machines, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 2:27:1–27:27.
[44] R. E. Shapire, Y. Singer, Improved Boosting Algorithms Using Confidence-Rated Predictions, Machine Learning 37 (3) (1999) 297–336.
[45] Y. Freund, R. E. Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.
[46] J. Friedman, T. Hastie, R. Tibshirani, Additive Logistic Regression: a Statistical View of Boosting, Tech. Rep. 12, Stanford University (Sep. 1998).
[47] M. Hagan, H. Demuth, M. Beale, Neural Network Design, PWS Publishing Company, 1996.
[48] M. Riedmiller, H. Braun, A Direct Adaptive Method for Faster Backpropagation Learning: the RPROP Algorithm, in: IEEE International Conference on Neural Networks, IEEE, 1993, pp. 586–591.